



Jouons avec React!

L'objectif des séances à venir est de réaliser une webapp en utilisant la librairie React.

Fonctionnalités attendues

L'application que vous avez à réaliser est une application mettant en oeuvre les concepts d'une application Rich Media : multimédia, interactive et synchronisée.

Elle devra proposer à l'utilisateur une interface lui permettant de visualiser un film, de naviguer dans le contenu par le biais d'un chapitrage mais également de consulter une carte illustrant les lieux où se déroule l'action.

L'utilisateur pourra interagir avec la carte en sélectionnant des points d'intérêt et ainsi accéder aux moments du film concernés.

Enfin une zone de discussion permettra à l'utilisateur de discuter avec d'autres utilisateurs et partager des pointeurs dirigeant vers des moments du film.####

Ce sujet de projet est commun avec le module Conception de Service & Ergonomie. Vous devrez donc intégrer les contraintes d'accessibilité à votre réalisation.

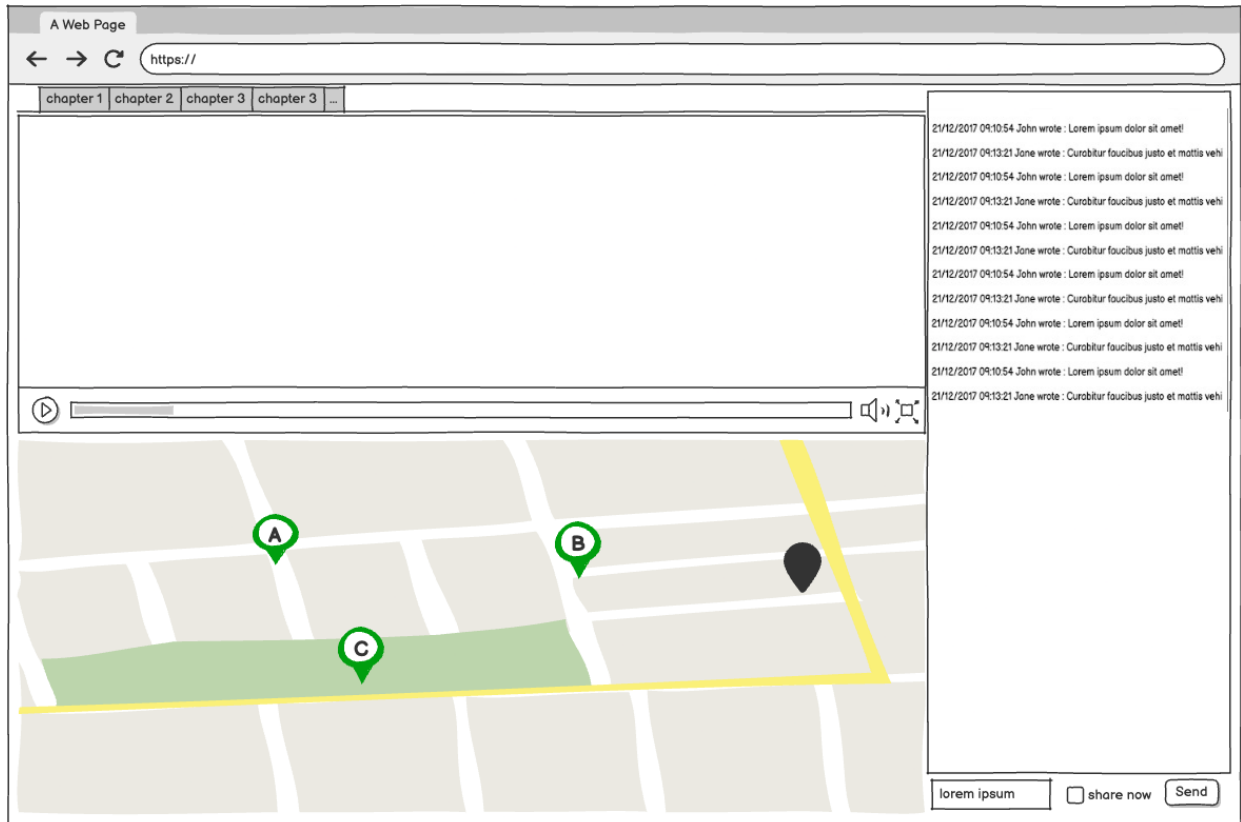
Pour ce faire vous aurez à implémenter:

- Un composant chargé de la lecture de la vidéo,
- Un composant proposant des options d'accessibilité tel que:
 - les sous-titres,
 - l'audio-description,
- un composant chargé d'afficher les chapitres de la vidéo, cliquer sur les chapitres

positionnera la vidéo au moment correspondant,

- un composant chargé d'afficher une carte avec des points d'intérêts liés à différents moments du film. Certains points d'intérêt sont liés à des scènes du film. Cliquer sur un de ces POI fera naviguer la vidéo à la ou les scènes concernées.
- un composant chargé d'afficher la chatroom associé à la vidéo. Cette chatroom permet de partager des messages ainsi que des moments-clefs du film. En cliquant sur un moment-clef partagé l'utilisateur accédera au moment dans le film.

Votre interface pourrait ressembler à ceci par exemple:



Ressources disponibles

Un backend est à votre disposition pour réaliser l'exercice. Ce backend fournit deux choses:

- un flux JSON contenant les informations relatives au film à afficher,
- une chatroom accessible en WebSocket

Les URLs sont les suivantes:

- Fichier JSON : <https://tp-iai3.cleverapps.io/projet>
- Chat WebSocket : <wss://tp-iai3.cleverapps.io>

Format de données

Le fichier film JSON

Le fichier JSON comporte les données suivantes:

- un noeud `film` contenant trois champs, `file_url`, `title`, `synopsis_url`, décrivant les informations sur le titre, l'url du fichier ainsi qu'une URL donnant son synopsis,
- un noeud `chapters` contenant l'URL du fichier JSON contenant le chapitrage,
- un noeud `audio-description` contenant l'URL du fichier JSON contenant la description des scènes en anglais, français et espagnol,
- un noeud `poi` contenant l'URL du fichier JSON contenant les points d'intérêts,
- un noeud `subtitles` contenant l'URL des fichier SRT contenant les sous-titres en anglais, français et espagnol.

Un exemple du fichier:

```
{
  "film" : {
    "file_url" : "https://upload.wikimedia.org/wikipedia/commons/2/24/Night",
    "title" : "Night of the Living Dead",
    "synopsis_url" : "https://en.wikipedia.org/wiki/Night_of_the_Living_Death",
  },
  "subtitles": {
    "en" : "https://tp-iai3.cleverapps.io/projet/subtitles-en.srt",
    "fr" : "https://tp-iai3.cleverapps.io/projet/subtitles-fr.srt",
    "es" : "https://tp-iai3.cleverapps.io/projet/subtitles-es.srt"
  }
},
  "audio-description": "https://tp-iai3.cleverapps.io/projet/description.json",
  "chapters": "https://tp-iai3.cleverapps.io/projet/chapters.json",
  "poi": "https://tp-iai3.cleverapps.io/projet/poi.json"
}
```

Le fichier chapters JSON

Le fichier JSON comporte un tableau contenant les chapitres. Chaque chapitre contient les données suivantes:

- un champ `chapter` contenant le numéro du chapitre,
- un champ `timestamp` contenant l'horodatage du début du chapitre,
- un champ `title` contenant le titre du chapitre en anglais,
- un champ `title_fr` contenant le titre du chapitre en français,
- un champ `title_es` contenant le titre du chapitre en espagnol,
- un champ `description` contenant un descriptif succinct du chapitre en anglais,
- un champ `description_fr` contenant un descriptif succinct du chapitre en français,
- un champ `description_es` contenant un descriptif succinct du chapitre en espagnol.

Un exemple du fichier:

```
[
  {
    "chapter": 1,
    "timestamp": "00:00:00",
    "title": "The Cemetery",
    "title_fr": "Le Cimetière",
    "title_es": "El Cementerio",
    "description": "Barbara and Johnny visit their father's grave in a remote cemetery.",
    "description_fr": "Barbara et Johnny visitent la tombe de leur père dans un cimetière isolé.",
    "description_es": "Barbara y Johnny visitan la tumba de su padre en un cementerio remoto.",
  },
  (...)
]
```

Le fichier audio-description JSON

Le fichier JSON comporte un tableau contenant les descriptions des scènes. Chaque scène contient les données suivantes:

- un champ `scene` contenant le numéro de la scène,
- un champ `timestamp` contenant l'horodatage du début de la scène,
- un champ `description` contenant un descriptif de la scène en anglais,
- un champ `description_fr` contenant un descriptif de la scène en français,
- un champ `description_es` contenant un descriptif de la scène en espagnol.

Un exemple du fichier:

```
[
  {
    "scene": 1,
    "timestamp": "00:00:00",
    "description": "Black and white film. Opening credits appear over a wide shot of a cemetery.",
    "description_fr": "Film en noir et blanc. Le générique d'ouverture apparaît sur une large vue d'un cimetière.",
    "description_es": "Película en blanco y negro. Los créditos de apertura aparecen sobre una amplia vista de un cementerio.",
  },
  (...)
]
```

Le fichier poi JSON

Le fichier JSON comporte un tableau contenant les points d'intérêts associés à différentes scènes du film. Chaque POI contient les données suivantes:

- un champ `id` contenant le numéro du point d'intérêt,

- un champ `title` contenant le nom du point d'intérêt en anglais,
- un champ `title_fr` contenant le nom du point d'intérêt en français,
- un champ `title_es` contenant le nom du point d'intérêt en espagnol,
- un champ `description` contenant un descriptif de la scène en anglais,
- un champ `description_fr` contenant un descriptif de la scène en français,
- un champ `description_es` contenant un descriptif de la scène en espagnol.
- un champ `timestamps` optionnel contenant un tableau des scènes liées au point d'intérêt. Chaque scène comporte les champs suivants:
 - un champ `time` avec l'horodatage de la scène,
 - un champ `scene` avec un bref descriptif de la scène en anglais,
 - un champ `scene_fr` avec un bref descriptif de la scène en français,
 - un champ `scene_es` avec un bref descriptif de la scène en espagnol,

Un exemple du fichier:

```
[
  {
    "id": 1,
    "title": "Evans City Cemetery",
    "title_fr": "Cimetière d'Evans City",
    "title_es": "Cementerio de Evans City",
    "latitude": 40.7681,
    "longitude": -80.0634,
    "description": "The cemetery where the opening scene was filmed, where",
    "description_fr": "Le cimetière où la scène d'ouverture a été filmée",
    "description_es": "El cementerio donde se filmó la escena de apertura",
    "timestamps": [
      {
        "time": "00:00:00",
        "scene": "Opening credits and arrival at cemetery",
        "scene_fr": "Générique d'ouverture et arrivée au cimetière",
        "scene_es": "Créditos de apertura y llegada al cementerio"
      },
      (...)
    ]
  },
  (...)
]
```

Les fichier sous-titres SRT

Les fichiers contiennent les sous-titres du film en anglais, français et espagnol au format SRT.

Le format SRT est un format utilisé pour les sous-titrages. Vous trouverez plus d'information

[ici]:<https://lokalise.com/blog/what-is-an-srt-file-subtitle-format-explained/>

Un exemple du fichier:

```
1
00:02:57,427 --> 00:03:00,726
They ought to make the day the time changes
the first day of summer.

2
00:03:00,806 --> 00:03:03,525
- What?
- Well, it's 8 o'clock and it's still light.

3
00:03:04,142 --> 00:03:06,732
A lot of good the extra daylight does us.

(...)
```

Le chat

En vous connectant au serveur de WebSockets vous recevrez dès la connexion un tableau contenant un ou plusieurs objets correspondant aux messages émis sur le chat avant que vous soyez connectés. Ces objets disposent de trois champs obligatoires, `when`, `name` et `message` ainsi qu'un champ optionnel `moment`. `when` contient le timestamp Unix du moment où le message a été reçu par le serveur, `name` contient le nom de son auteur et `message` le contenu. `moment` contient le timestamp en secondes du moment du film partagé.

Un exemple de réponse du serveur de Websockets:

```
[
  {
    "when": "1580742794",
    "name": "Alice",
    "message": "Hi, I'm Alice!"
  },
  {
    "when": "1580742479",
    "name": "Bob",
    "message": "Hi, I'm Bob. Checkout this moment!",
    "moment": 462
  },
  (...)
]
```

Critères d'évaluation

La note que vous aurez à l'issue du module est composée:

- à 50% de la note obtenue lors du devoir sur table portant sur le cours d'introductions à l'évolution du développement d'interfaces RIA ainsi que sur l'initiation à React,
- à 50% de la note obtenue avec le projet dont vous lisez le sujet.

Ce nano-projet, **qui n'est à réaliser que pendant le temps où vous êtes en cours**, est évalué sur les critères suivants:

- code doit obligatoirement impérativement compiler après un `git clone` suivi d'un `npm install & npm start`. Cette condition vous garanti un minimum de 10/20 sur la note du projet.
- présence obligatoire des composants listés au point ci-dessus
- votre application doit remplir les conditions d'une RIA : interactive, multimédia et synchronisée. Les différents composants présentent tous des caractéristiques qui peuvent influencer les autres : cliquer sur un point d'intérêt sur la carte change le positionnement de la vidéo ainsi que les mots-clefs affichés. Cliquer sur un chapitre avance ou recule la lecture de la vidéo, change le sous-titrage et l'audio-description,...
- style visuel de l'application. L'esthétique est libre mais votre code doit mettre en évidence l'usage de CSS.

Afin que je puisse évaluer correctement votre travail vous devrez le partager sur un dépôt git de votre choix et m'y donner accès en précisant les noms et prénoms des membres de l'équipe (dans le README de votre dépôt par exemple). Afin d'évaluer la répartition du travail dans les membres de l'équipe prenez soin de pousser vos contributions respectives régulièrement sur le dépôt. Si vous travaillez en pair programming, précisez-le dans votre README. J'utilise git blame afin d'obtenir une estimation de la contribution de chaque membre du projet, aussi est-il important que chaque membre de l'équipe contribue au dépôt git.

Mon adresse mail pour me partager l'accès à vos dépôts : marc.poppleton@etik.com

Aide

Afin de vous aider à implémenter les différentes fonctionnalités vous trouverez ci-dessous des liens vers des librairies qui peuvent vous être utiles.

Librairies utiles

- Pour intégrer des cartes OpenstreetMap, vous pouvez par exempl utiliser la librairie [React-leaflet](#)

- Pour intégrer des composants déjà stylisés, vous pouvez par exemple utiliser la librairie [Bootstrap React](#)
- Pour intégrer un lecteur video, vous pouvez utiliser le player de la librairie [React Player](#)

Il s'agit de suggestions, vous êtes totalement libre d'utiliser d'autres librairies ou de créer les votre!

Attention

Pour que votre carte [React-leaflet](#) s'affiche correctement vous devez non seulement importer les dépendances comme précisé dans la [documentation](#) mais également importer la feuille de style CSS et définir les attributs de hauteur et largeur dans votre App.css.

```
@import url("https://unpkg.com/leaflet@1.6.0/dist/leaflet.css");

#id_de_votre_composant_map{
  width:600px;
  height:400px
}
```